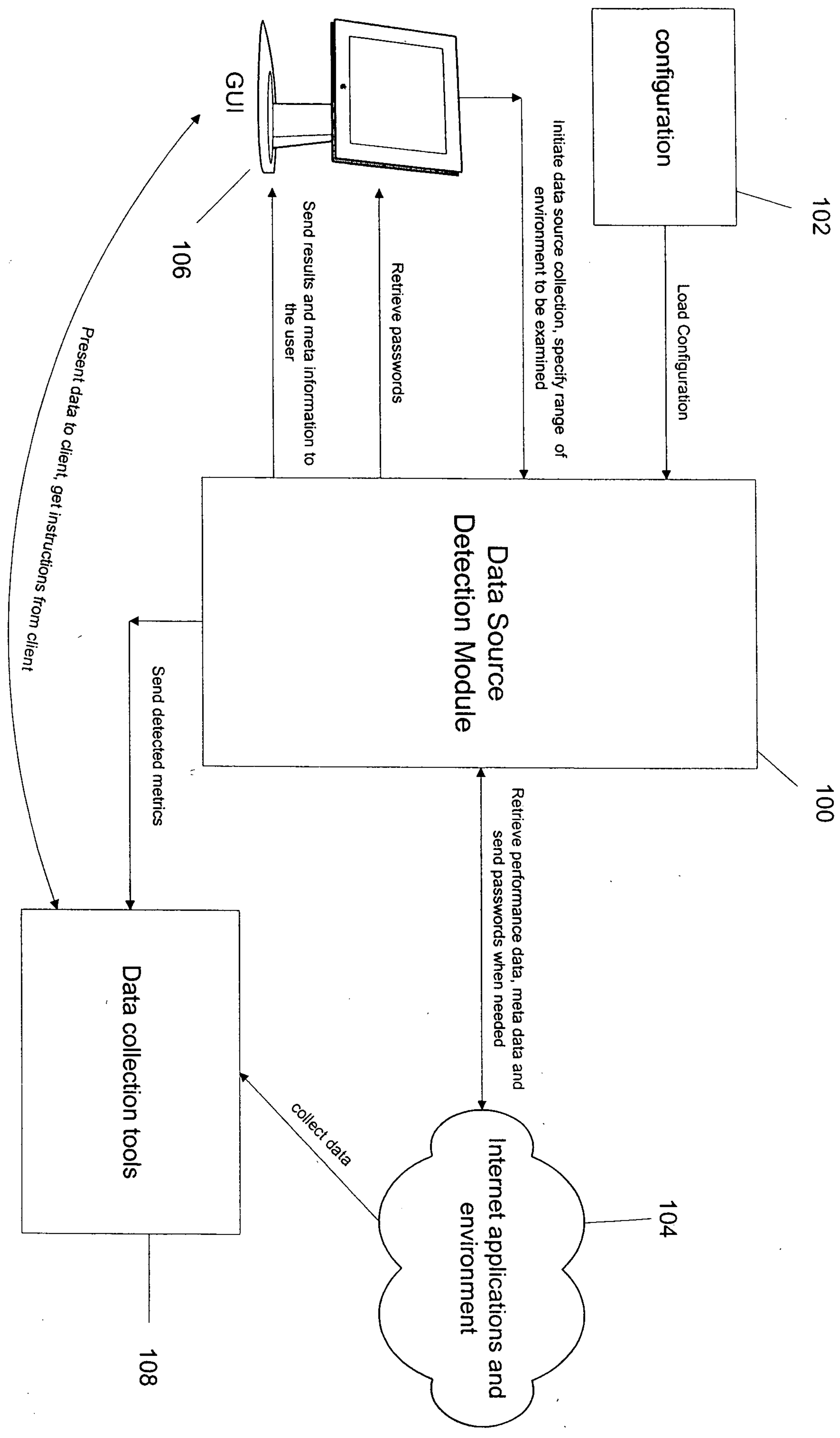

configuration
102
Load Configuration
Initiate data source collection, specify range of environment to be examined
Retrieve passwords
Send results and meta information to the user
GUI
106
Data Source Detection Module
100
Retrieve performance data, meta data and send passwords when needed
Send detected metrics
Present data to client, get instructions from client
Data collection tools
108
collect data
Internet applications and environment
104

Fig. 1

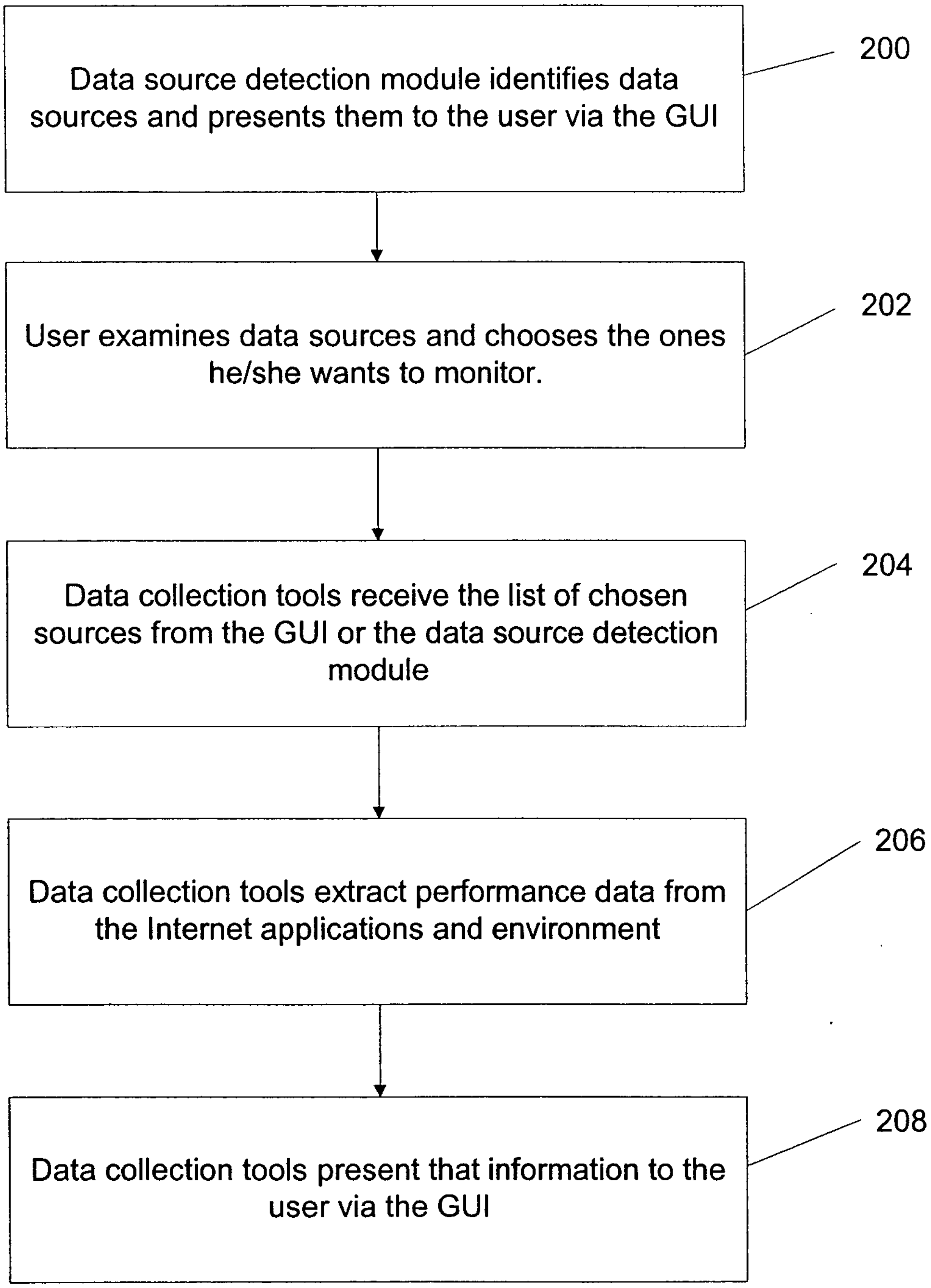

Data source detection module identifies data sources and presents them to the user via the GUI
200
User examines data sources and chooses the ones he/she wants to monitor.
202
Data collection tools receive the list of chosen sources from the GUI or the data source detection module
204
Data collection tools extract performance data from the Internet applications and environment
206
Data collection tools present that information to the user via the GUI
208

Fig 2

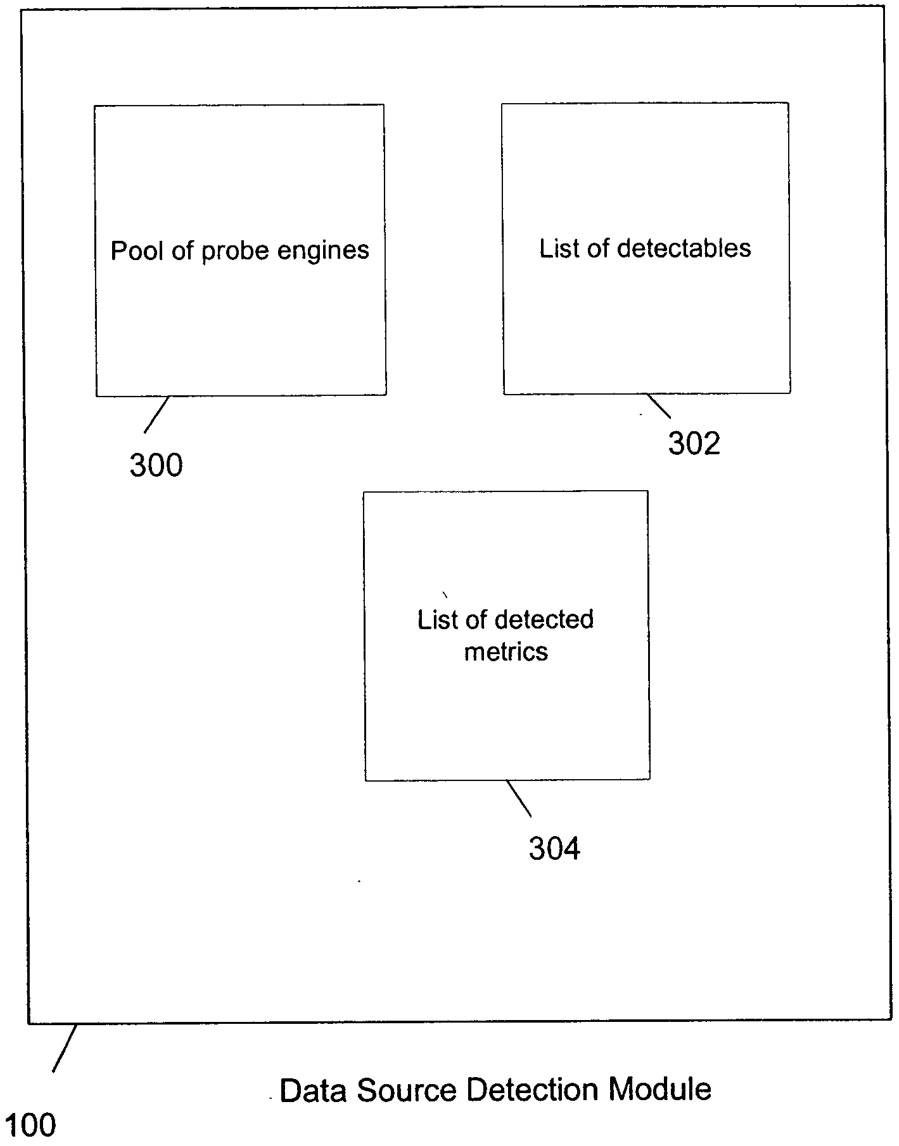

Pool of probe engines
300
List of detectables
302
List of detected metrics
304
100
Data Source Detection Module

Fig. 3

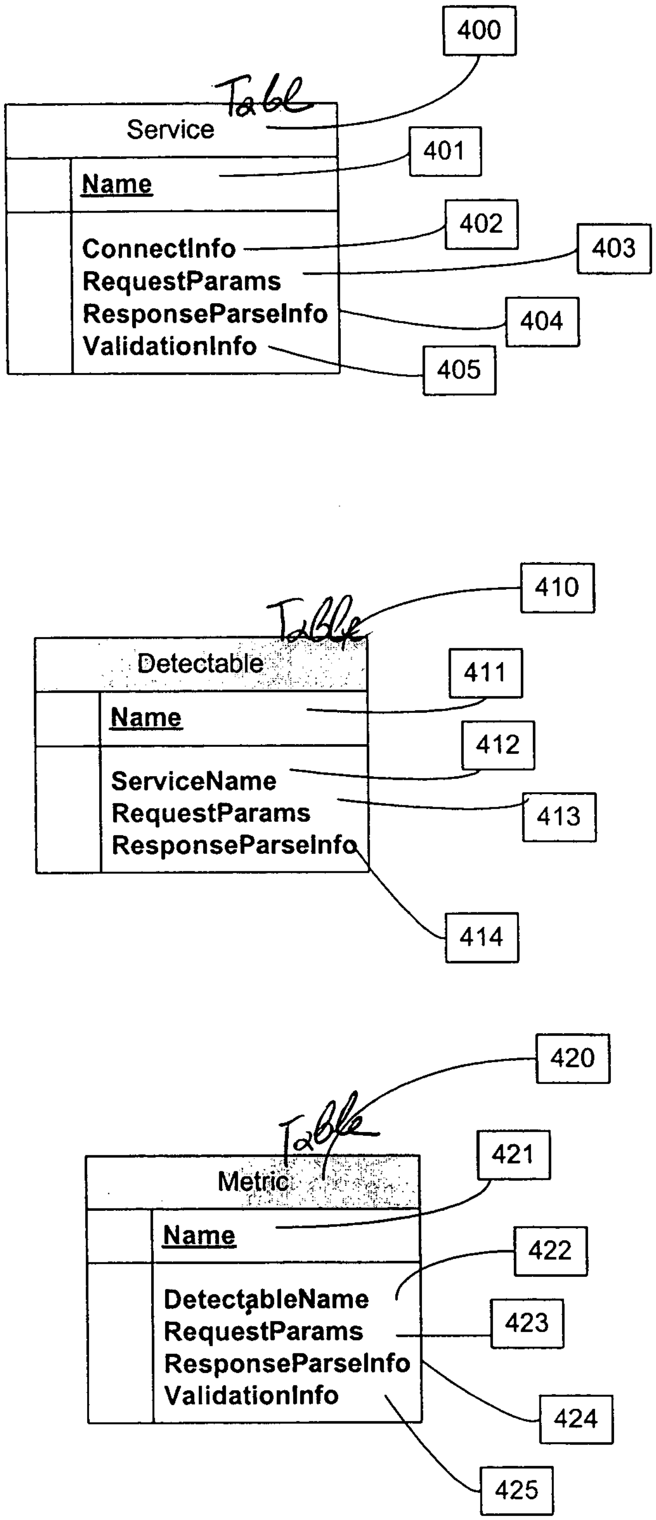

400
Table
Service
401
Name
402
ConnectInfo
403
RequestParams
404
ResponseParseInfo
405
ValidationInfo
410
Table
Detectable
411
Name
412
ServiceName
413
RequestParams
ResponseParseInfo
414
420
Table
Metric
421
Name
422
DetectableName
423
RequestParams
ResponseParseInfo
424
ValidationInfo
425

Fig.4

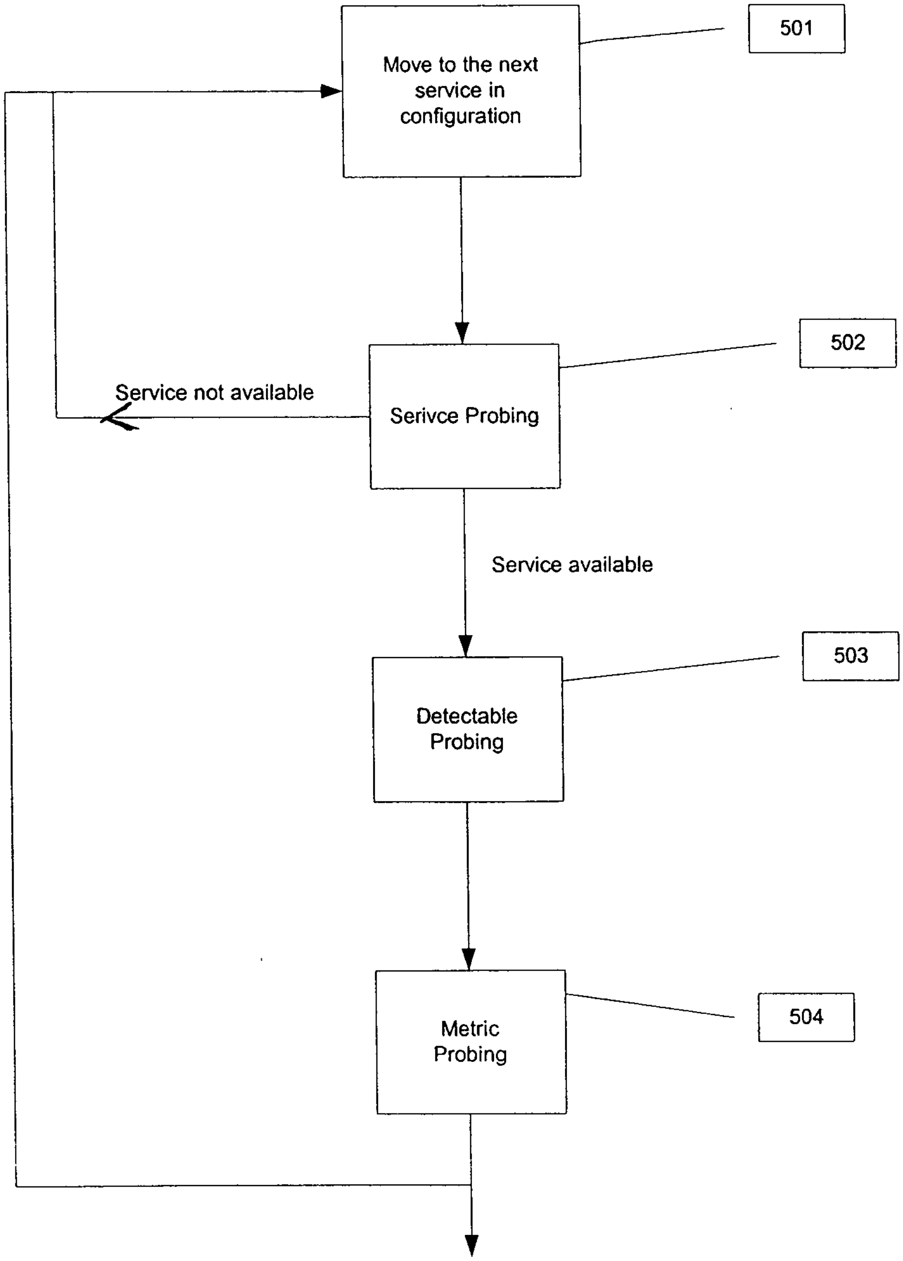

Move to the next service in configuration
501
Serivce Probing
502
Service not available
Service available
Detectable Probing
503
Metric Probing
504

Fig.5

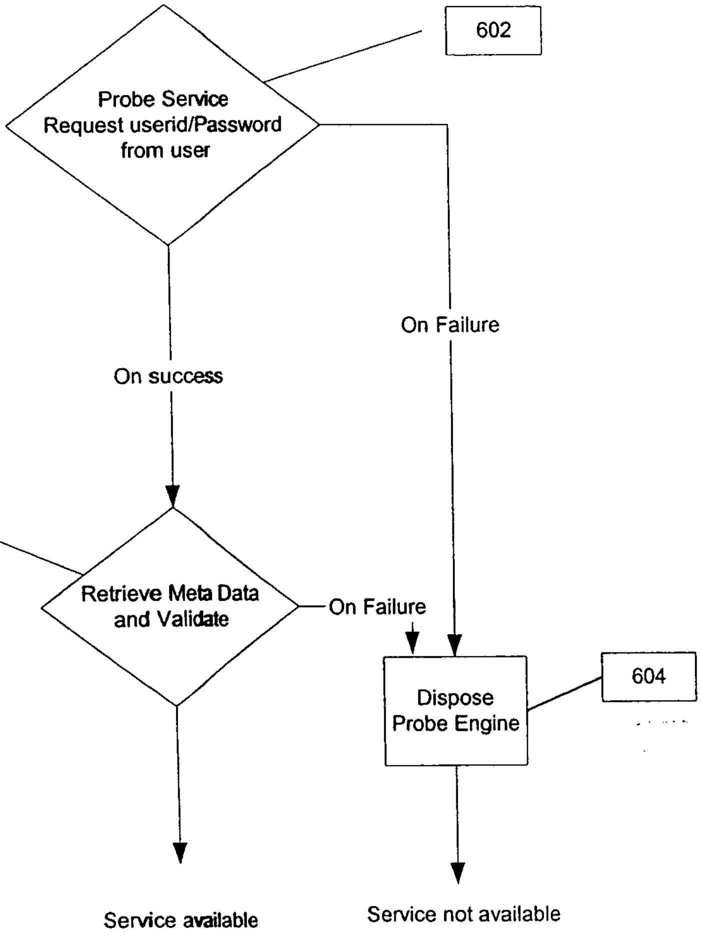

602
Probe Service
Request userid/Password
from user
On Failure
On success
603
Retrieve Meta Data
and Validate
On Failure
Dispose
Probe Engine
604
Service available
Service not available

Fig.6

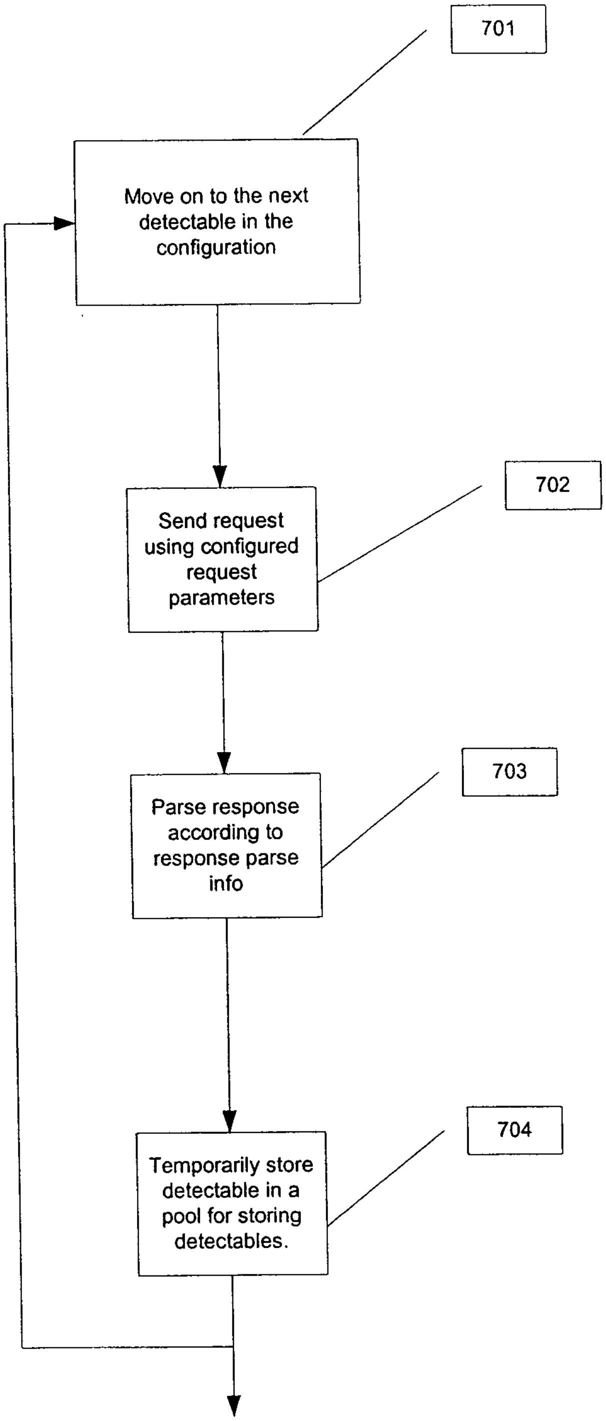

701
Move on to the next detectable in the configuration
702
Send request using configured request parameters
703
Parse response according to response parse info
704
Temporarily store detectable in a pool for storing detectables.

Fig.7

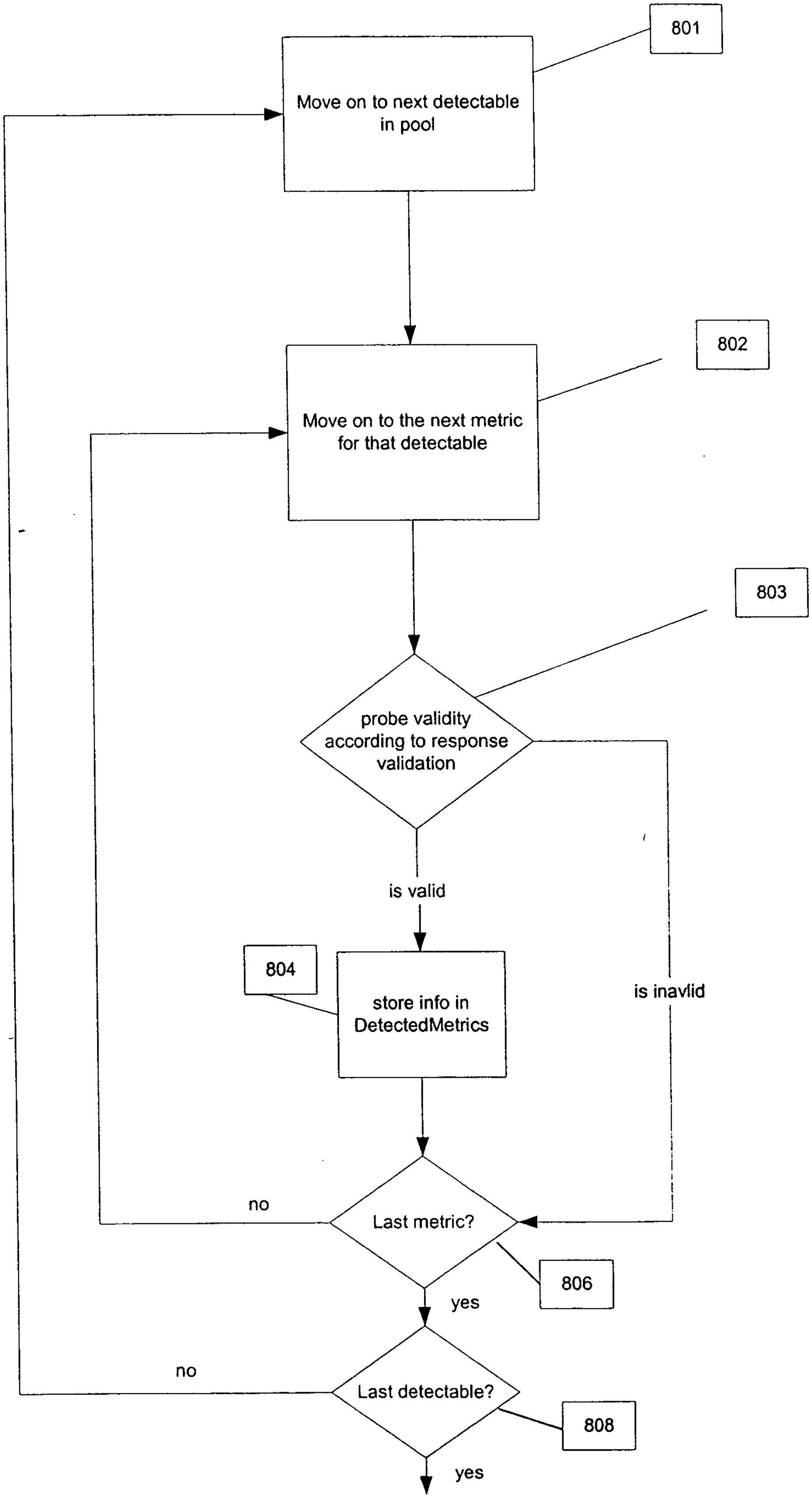

801
Move on to next detectable in pool
802
Move on to the next metric for that detectable
803
probe validity according to response validation
is valid
is inavlid
804
store info in DetectedMetrics
no
Last metric?
806
yes
no
Last detectable?
808
yes

Fig.8

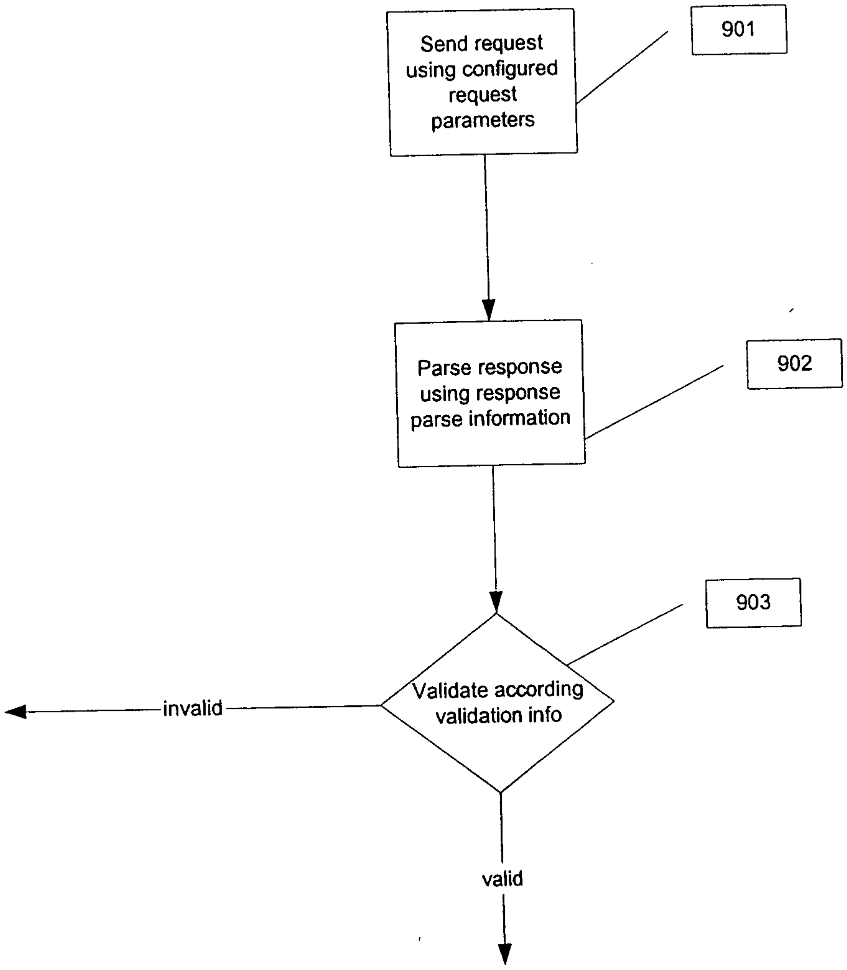

Send request using configured request parameters
901
Parse response using response parse information
902
903
Validate according validation info
invalid
valid

Fig.9

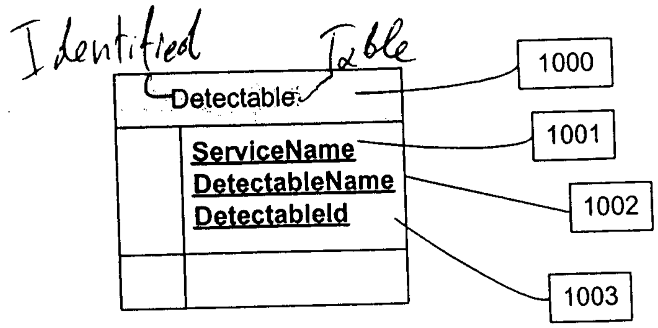

Identified
Table
Detectable
ServiceName
DetectableName
DetectableId
1000
1001
1002
1003

Fig.10

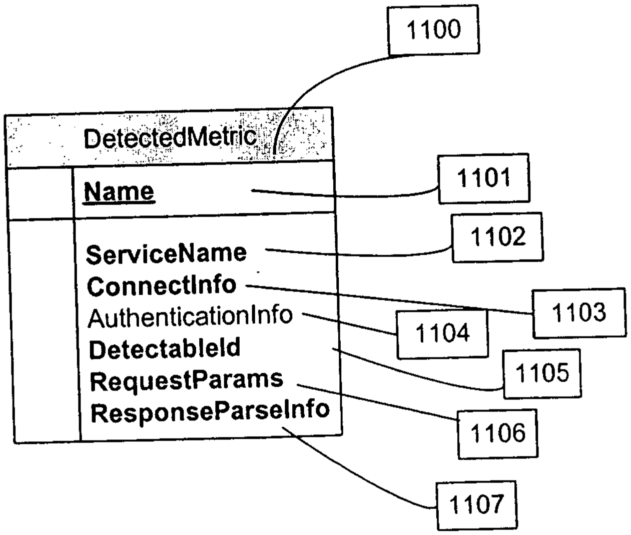
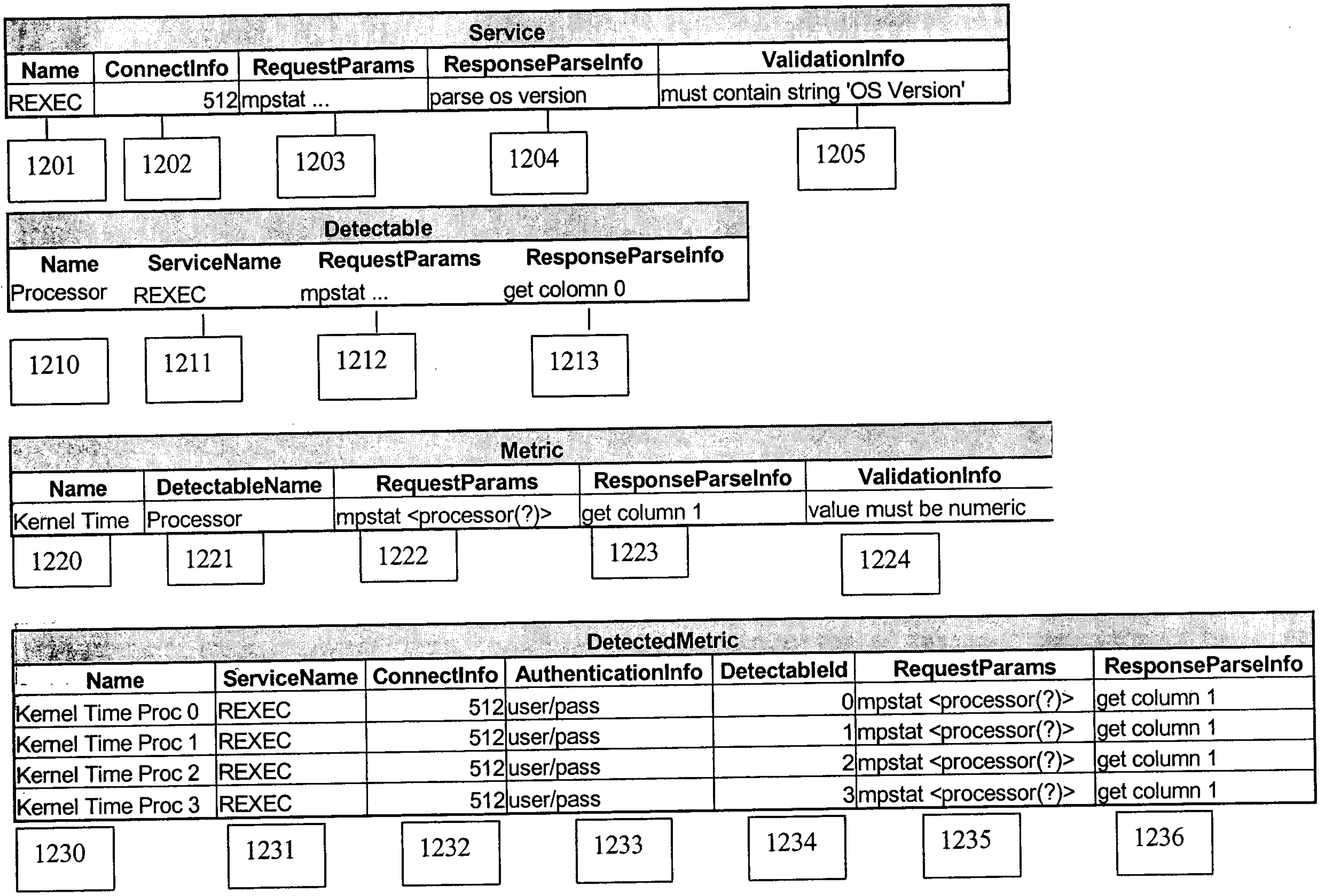

DetectedMetric
Name
ServiceName
ConnectInfo
AuthenticationInfo
DetectableId
RequestParams
ResponseParseInfo
1100
1101
1102
1103
1104
1105
1106
1107

Fig.11


1201
1202
1203
1204
1205
1210
1211
1212
1213
1220
1221
1222
1223
1224
1230
1231
1232
1233
1234
1235
1236

**Service**

| Name | ConnectInfo | RequestParams | ResponseParseInfo | ValidationInfo |
|---|---|---|---|---|
| REXEC | 512 | mpstat ... | parse os version | must contain string 'OS Version' |

**Detectable**

| Name | ServiceName | RequestParams | ResponseParseInfo |
|---|---|---|---|
| Processor | REXEC | mpstat ... | get colomn 0 |

**Metric**

| Name | DetectableName | RequestParams | ResponseParseInfo | ValidationInfo |
|---|---|---|---|---|
| Kernel Time | Processor | mpstat <processor(?)> | get column 1 | value must be numeric |

**DetectedMetric**

| Name | ServiceName | ConnectInfo | AuthenticationInfo | DetectableId | RequestParams | ResponseParseInfo |
|---|---|---|---|---|---|---|
| Kernel Time Proc 0 | REXEC | 512 | user/pass | 0 | mpstat <processor(?)> | get column 1 |
| Kernel Time Proc 1 | REXEC | 512 | user/pass | 1 | mpstat <processor(?)> | get column 1 |
| Kernel Time Proc 2 | REXEC | 512 | user/pass | 2 | mpstat <processor(?)> | get column 1 |
| Kernel Time Proc 3 | REXEC | 512 | user/pass | 3 | mpstat <processor(?)> | get column 1 |

**Fig. 12**